Master's Project Presentation

Sam Barrett, 1803086

University of Birmingham

October 30, 2020

My Topic

Applications of Genetic Algorithms on Fully Autonomous Road Networks

- Semi-autonomous vehicles are becoming more prevalent
- Roads are becoming more congested with a 78% increase in motor traffic since 1993 [3]
- Fully autonomous vehicle trials have been legal in parts of the US since 2015[1], with the UK set to follow by next year (2021)[4]
- Much of the current research into autonomous vehicle routing focuses on environments where human drivers are still present
- By removing the human element and working on theoretical *fully autonomous road networks* we can make many useful assumptions about the behaviour of other vehicles
- The solution to road congestion is not to build bigger roads, it is to optimise the traffic flows.
- Just 78.2% of journeys on the UK Highway Agencies roads were *on time* in the year ending June 2014 [6]

- I am choosing to focus on the applications of Genetic Algorithms on Theoretical Fully autonomous road networks, with a view to extend into the possible applications of quantum computers on the field in the future.
- I feel the advent of fully autonomous road networks is a logical next step in making roads safer and more efficient through the use of technology.
- Fully autonomous driving trials have been legal in parts of the states for years with the UK following soon.
- Most research and all currently implemented systems focus on semi-autonomous environments whereby self-driving vehicles and humans co-exist on shared roads.
- I propose it is both safer, easier and more efficient to implement, fully autonomous road networks where humans are not able to operate their vehicles.
- In such a system, sensor data would be shared between all vehicles near instantaneously allowing for much faster and less-selfish route planning, leading to net decreases in travel time.

Literature Review I

I am currently intending to pursue my research assuming the absence of classical speed lanes as described by Kala and Warwick in [5].
I have chosen to focus on the applications of Genetic Algorithms on the field for 3 reasons:

1. It is a class of optimisation algorithms that I find particularly interesting
2. GAs are *probabilistically optimal and complete*, i.e given infinite time, they will always produce the global optimal solution if such a solution exists
3. It is a class of algorithm that has seen relatively minimal research in my the specific sub-area

- As previously mentioned most current research into GA applications within the car industry has a very broad scope.
- Designing possible solutions that would fit into the current road networks easily.
- I am intending to focus on a much more aspirational system, specifically looking at theoretical autonomous Motorways.
- This enables me to overhaul the current road layout which was designed to aid human drivers not the overall efficiency of the system.

I have chosen to focus on GAs as opposed to other possible AI veins for a few reasons.

- One personal reason is that I find them particularly interesting.
- One more concrete reason is that they have the very useful property of being both probabilistically optimal and probabilistically complete. Meaning that given infinite time they not only will find *a* solution but they will find **the** optimal solution.
- And finally they have seen relatively minimal research in the field of vehicle planning with the limelight being taken by technologies such as Deep Learning or Reinforcement Learning

Literature Review II

- Other approaches involve *black box approaches*.
- The downside of such an approach is that it is very difficult to reason and predict the actions of the system. The ability to assure safety of such a critical system is very important and so GAs offer a much more predictable result
- Kala and Warwick [5] proposed a system of two coordinate systems to safely represent points on the road within Cartesian space.

- A many of the technologies being researched with regards to vehicular planning suffer from the same problem from my point of view, they are *black box approaches* meaning given a model, it is very difficult, if not impossible, to reason about and predict the decisions it makes. Such systems could end up making decisions based on imperfections in the training set. Such issues will make these systems less dependable and make people less likely to put their faith in them.
- In a paper from 2013, Kala and Warwick proposed a method of representing roads as a set of boundary functions in Cartesian space. All points on the road are defined using these functions as a new basis, this seems to be a good approach as it eliminates the possibility of plotting routes outside of the road space.

Not final but preliminary implementations have used Julia[2]

- C-like performance
- Python & Matlab -like syntax
- Matlab like matrices
- Allows for both OO and functional approaches to problems
- Can be compiled
- Allows for use of Unicode in variable & function names so implementations of advanced mathematical expressions are much more readable

Talk about what criteria an implementation would need to satisfy, talk about methods to satisfy them and condense lang choice to focus on fulfillment of criteria

```julia
function Σ(xs::Array{Int64})
    ret = 0
    for x in xs
        ret += x
    end
    ret
end

xs = [1,2,3]

Σ(xs) == reduce(+,xs) # true
```

Figure: Example Julia code

Alternative languages include C, Python and Rust

Alternative languages include C, Python and Rust

C:

- Compiles down to binary
- Antiquated syntax
- Possible (and easy) to write memory unsafe code
- Vast array of libraries due to age & use
- No functional properties, harder to implement readable mathematics

Alternative languages include C, Python and Rust

C:

- Compiles down to binary
- Antiquated syntax
- Possible (and easy) to write memory unsafe code
- Vast array of libraries due to age & use
- No functional properties, harder to implement readable mathematics

Python:

- Simple syntax
- Wealth of stress-tested libraries
- Slow relative to alternatives
- unable to compile to binary format
- Has some functional capabilities
- Has some static typing ability

Alternative languages include C, Python and Rust

C:

- Compiles down to binary
- Antiquated syntax
- Possible (and easy) to write memory unsafe code
- Vast array of libraries due to age & use
- No functional properties, harder to implement readable mathematics

Python:

- Simple syntax
- Wealth of stress-tested libraries
- Slow relative to alternatives
- unable to compile to binary format
- Has some functional capabilities
- Has some static typing ability

Rust:

- Slower to prototype in as stricter type system to guarantee memory safety
- Memory safe, advantage over C/C++
- Very performant, runs well on embedded systems
- Relatively large binaries due to static dependency linking
- Easier to package & deploy than Julia

References I

- Autonomous Vehicles — Self-Driving Vehicles Enacted Legislation.
  https://www.ncsl.org/research/transportation/autonomous-vehicles-self-driving-vehicles-enacted-legislation.aspx.
- The Julia Programming Language.
  https://julialang.org/.
- Highways England network journey time and traffic flow data.
  https://data.gov.uk/dataset/9562c512-4a0b-45ee-b6ad-afc0f99b841f/highways-england-network-journey-time-and-traffic-flow-data, June 2015.
- .
  UK wants fully autonomous cars on road.
  BBC News, Feb. 2019.

References II

R. Kala and K. Warwick.
Motion planning of autonomous vehicles in a non-autonomous vehicle environment without speed lanes.
Engineering Applications of Artificial Intelligence, 26(5):1588–1601, May 2013.

I. A. roads travel time measures 020 7944 3095.
Reliability of journeys on Highways Agency roads, England: April to June 2014.
https://www.gov.uk/government/statistics/reliability-of-journeys-on-highways-agency-roads-england-apr-to-jun-2014.